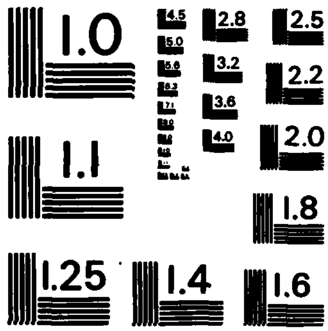
AD-A161 871 AIDS IN VALIDATING A CONTRACTOR'S COST ESTIMATE(U) 1/1
DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA
B H RUDWICK 1985

UNCLASSIFIED F/G 5/1 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(1)

Aids In Validating A Contractor's Cost Estimate

by

Bernard H. Rudwick, P.E.

Professor of Financial Management
Defense Systems Management College
Fort Belvoir, VA

Presented at the
19th Annual Department of Defense
Cost Analysis Symposium
Xerox Training Center
Leesburg, Virginia
September 17 - 20 1985

AIDS IN VALIDATING A CONTRACTOR'S COST ESTIMATE

Bernard H. Rudwick, P.E.
Professor of Financial Management
Defense Systems Management College
Fort Belvoir, Virginia

INTRODUCTION

Part of any evaluation of a contractor's system proposal is the evaluation of contractor's cost estimate to validate that the contractor's proposed cost is "fair and reasonable". Experience has shown that if this proposal involves a sole source negotiation with the contractor, the contractor's cost estimate tends to be quite conservative; i.e., high. Since the contractor desires to allow himself a factor of safety, a conservative cost estimate also tends to increase profit. On the other hand, if there is competition for a cost-reimbursable contract, the contractor's cost estimate may be unreasonably low. In both cases, it is to the government's advantage to obtain some independent estimate of the expected cost so that proper management decisions and program control may be achieved.

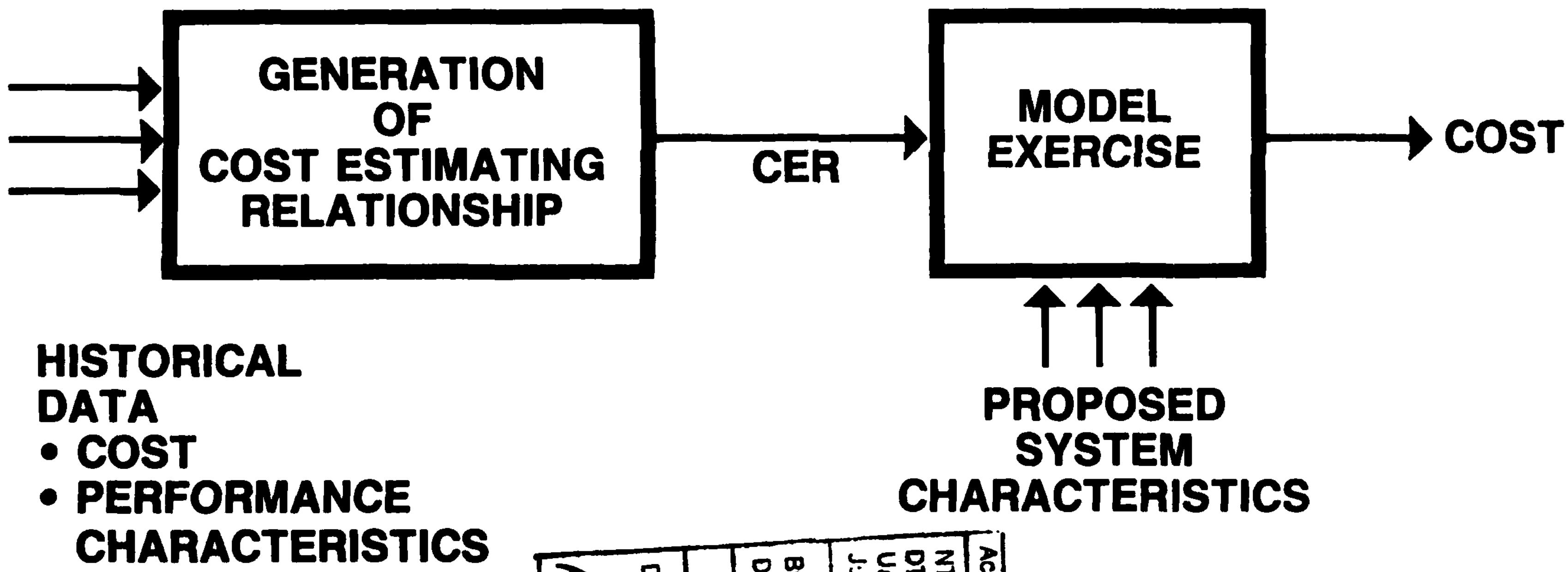
Figure 1 illustrates the generic approach used to generate a cost estimate in a systematic fashion. The cost analyst first needs to collect a data base linking some characteristic or set of characteristics of the type of system under consideration to the same characteristic(s) and cost of similar systems acquired in the past. From this data one or more cost estimating relationships (CER) are generated using one or more of the four cost estimating methods most appropriate to the problem: 1) Engineering bottoms-up or grass roots method; 2) analogy method; 3) extrapolation from actuals method (a form of analogy); 4) parametric method. An appropriate set of input characteristic values describing the proposed system is then used as the input to the CER, and the final cost is calculated using some analytical or simulation method.

Once the contractor's cost estimate is received by the government, its accuracy may be validated using one of the following methods. One approach is for the validator to generate an independent cost estimate using his or her own data base and input values. If the independent estimate is resonably close to the contractor's submitted estimate, validation is achieved. The difficulty with this approach is that it requires the government cost estimator to maintain a sufficiently large enough data base to generate an accurate enough cost estimating relationship (CER), and to have access to government experts who can estimate the proper input values required to use the CER.

A second approach is to perform a "Should Cost" analysis. This approach analyzes the work process which the contractor has proposed, looking for improvements which can be made which will result in higher efficiencies; i.e., reduced costs. The difficulty with this approach is the amount of effort required to understand and analyze the contractor's work process, and the ability of the government experts to generate improvements in the process which the contractor will accept.

FIGURE 1

GENERATING A COST ESTIMATE



A third approach is for the government cost analyst to audit the original analysis submitted by the contractor. Unfortunately many times the entire analysis performed in generating the cost estimate, including the "credible evidence" which supports the analysis, is not submitted to the government. Hence it is difficult for the government to audit the estimate.

Obviously, performing some combination of these three approaches will yield greater insight into what the final cost will be, but this requires more effort than using any single approach.

This paper will concentrate on the third method by indicating what data the contractor should provide as "credible evidence" to support the accuracy of his cost estimate. This will be done in the following way. First, by listing a set of six key questions which the contractor should answer when generating his cost estimate. Second, by illustrating the form that the answers should take.

SIX KEY QUESTIONS

As indicated previously, there are six key management questions which a reviewer should have the cost analyst address when generating a cost estimate. These questions, listed in Table 1, relate to the method used in developing the cost estimate. These questions should be stated in the Request for Proposal (RFP) so that the cost analyst will gather and submit appropriate back-up data to support his cost estimate. A discussion of these questions and a summary of the type of answers required now follows. The next section provides detailed answers to these questions with respect to a specific software cost estimating model used as an example to illustrate the recommended approach.

Question #1. What cost estimating methodology was used?

Answer: Four types of cost estimating methodologies are generally used, the choice depending on the amount of data available. These are: 1) Engineering or Bottoms-up; 2) Analogy; 3) Parametric; or 4) Extrapolation from Actuals. The specific type of method employed in generating the estimate should be stated.

Question #2. What cost estimating equations were used?

Answer: Most cost estimators use cost estimating equations (called Cost Estimating Relationships or CERs) which estimate the system cost as a function of a set of input characteristics. The specific equations (or algorithms) used in the CER should be stated.

Question #3. How was the CER derived and what is its uncertainty?

Answer: Most CERs are derived by generating some mathematical equation which best fits a set of data from similar type systems which have been developed or produced in the past. Since the fit of the equation to the points is rarely perfect, the use of the equation results in an estimate which is also imperfect. However, statistical methods exist which enable us to quantify the amount of uncertainty in the equation which was fitted to the data points. In validating this statistical analysis, the validator is concerned with two main points. First, what data points were used, so

SIX KEY QUESTIONS REVIEWER SHOULD ASK

- WHAT COST ESTIMATING METHOD WAS USED?
- WHAT COST ESTIMATING EQUATIONS WERE USED?
- HOW WAS CER DERIVED AND WHAT IS ITS AMOUNT OF UNCERTAINTY?
- WHAT INPUT VALUES WERE USED AND WHAT WAS THE RANGE OF UNCERTAINTY FOR EACH INPUT?
- WHAT WAS THE IMPACT OF UNCERTAINTIES ON CER OUTPUT?
- HOW WELL DOES THE CER CORRESPOND TO MY SYSTEM OR HOW WAS IT MADE TO CORRESPOND?

that the appropriateness of these past systems to the proposed system can be validated. Second, how good a fit is the derived equation to the set of data points? Measures of the model's accuracy such as the correlation coefficient (R), coefficient of determination (R^2), and Standard Error of Estimate (SEE) provide this answer.

Question #4. What input values were used and what is the range of uncertainty associated with each of the input factors?

Answer: In this case the contractor's range of uncertainty for each input factor (as a three point estimate) should be reviewed and validated by the government reviewing team. If there are strong differences of opinion about any value, a new range of uncertainty should be formulated by the team so that the government's recalculation of cost can be generated if desired.

Question #5. How does the set of uncertainties in both the inputs and the CER effect the output cost estimate?

Answer: The set of uncertainties in the inputs and the CER can be translated into output uncertainties by the use of sensitivity analysis or probabilistic analysis.

Question #6. How well does the CER correspond to the particular system or how can the CER be made to correspond?

Answer: If the cost analyst generated his own CER from data be collected, comparing the particular system to the systems used in the data base will indicate the relevence of the CER. On the other hand, if someone else's CER were used, the CER should be calibrated to reflect the contractor's way of doing business.

APPLYING THIS APPROACH

Having been alerted in the RFP to the substantiating data required (the six key questions), here is an example of how these key questions should be answered by the contractor (or any analyst generating a cost estimate in a systematic fashion). The specific example used for illustration is the use of COCOMO, a software cost estimating model.

Question #1. What cost estimating methodology was used?

Answer: In our example, the Constructive Cost Model (COCOMO) uses the parametric cost estimating method.

Question #2. What cost estimating equations were used?

Answer: In his book, Boehm describes three COCOMO models (Basic, Intermediate, and Detailed). As might be expected, each provides increasing accuracy over its predecessor, but at the cost of additional analytical effort. This paper will concentrate on the Intermediate model.

Intermediate COCOMO Model Nominal Estimating Equation

To develop the COCOMO model, Boehm strove to assemble a historical data base which would be as uniform and homogenous as possible. To do this he analyzed a carefully screened set of 63 completed software projects. He then developed a set of characteristics which appeared to effect development cost (and schedule), and through interviews collected the values of these characteristics for each of the 63 projects.

The COCOMO intermediate model consists of two primary components. The first consists of a nominal estimating equation representing software development effort (in project man-months) as a function of the number of delivered source instructions (or lines of code). Obviously, the man-months of development effort can be readily converted into labor costs in dollars by multiplying man-months by the fully burdened labor rate per month. A similar equation is also available for schedule time (in months).

The numerical characteristics of these equations depend on which of three different modes of software development is to be used, as follows:

Mode	Nominal Man-Months	Schedule Time
Organic	$MM = 3.2\ (KDSI)^{1.05}$	$TDEV = 2.5\ (MM)^{.38}$
Semi-detached	$MM = 3.0\ (KDSI)^{1.12}$	$TDEV = 2.5\ (MM)^{.35}$
Embedded	$MM = 2.8\ (KDSI)^{1.20}$	$TDEV = 2.5\ (MM)^{.32}$

where: MM = man-months required to develop the software.
KDSI = number representing thousands of delivered source instructions.
TDEV = time to develop the software in months.

To select the appropriate COCOMO equation the cost analyst must determine which mode best defines the project being estimated:

Mode	Project Characteristics
Organic:	In-house software development and relatively small development teams in a stable environment.
Embedded:	Software program operates within tight constraints. Software typically embedded in a complex hardware system and must often "take up slack" when difficulties are encountered.
Semi-detached:	Contains a mixture of organic and embedded mode characteristics. May contain some rigorous interfaces (tight constraints) and some very flexible interfaces. (The word semi-detached comes from this "partial" flexibility.) This category is in between the other two.

To achieve a uniform and homogenous data base, other key definitions and assumptions were followed and associated with these equations:

o The primary cost driver is the number of delivered source instructions (KDSI) developed for the project. This is defined as follows:

- Delivered. In essence, all software modules designed and developed from scratch or significantly rebuilt are included as delivered software. Generally, this term excludes nondelivered support software such as test drivers. However, if these are developed with the same care as delivered software, with their own reviews, test plans, documentation, etc., then they should also be counted.

- Source Instructions. COCOMO defines this term to include all delivered program instructions created by project personnel and processed into machine code by some combination of preprocessors, compilers, and assemblers. It excludes comment cards, unmodified software, and other control language, format statements, and data declarations. Instructions are defined as lines of code or card images. Thus a line containing two or more source statements counts as one instruction; a five-line data declaration counts as five instructions.

The development period covered by COCOMO cost estimates begins at the beginning of the product design phase (successful completion of a software requirements review) and ends at the end of the integration and test phase (successful completion of a software acceptance review). Costs and schedules of other phases are estimated separately.

The COCOMO cost estimates cover specific activities that are indicated on the typical software work breakdown structure (WBS). For example, the development estimate covers management and documentation efforts, but excludes some efforts which take place during the development period such as user training, installation planning, and conversion planning.

The COCOMO cost estimates cover all direct labor on the project for the activities indicated in the WBS. Thus they include program managers and program librarians, but exclude computer center operators, secretaries, higher management, janitors, and so on.

While these are the definitions which Boehm used in generating COCOMO, as long as the COCOMO model is calibrated to the contractor's method of operation, any consistent definition of source instructions and these other factors may be used, as will be described later.

COCOMO Development Effort Multipliers

Selecting the appropriate mode and using the COCOMO equations with KDSI as the independent variable will provide only a very rough cost estimate. To improve the estimate, the Intermediate COCOMO model uses 15 other cost driver attributes which are assigned values and factored in as multipliers to the basic equations. These cost drivers are presented in Table 2 along with their associated numerical multiplier values. (The Boehm reference has tables that provide criteria and assistance in assigning a value to each of the 15 cost drivers).

For example, what is the required software reliability? As shown in Table 2, five categories of reliability are defined, from very low to very high. If the required

Table 2. SOFTWARE DEVELOPMENT EFFORT MULTIPLIERS

Cost Drivers	Ratings: Very Low	Low	Normal	High	Very High	Extra High
Product Attributes						
RELY Required software reliability	.75	.88	1.00	1.15	1.40	
DATA Data base size		.94	1.00	1.08	1.16	
CPLX Product complexity	.70	.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME Execution time constraint			1.00	1.11	1.30	1.66
STOR Main storage constraint			1.00	1.06	1.21	1.56
VIRT Virtual machine *		.87	1.00	1.15	1.30	
TURN Computer turnaround time		.87	1.00	1.07	1.15	
Personnel Attributes						
ACAP Analyst capability	1.46	1.19	1.00	.86	.71	
AEXP Applications experience	1.29	1.13	1.00	.91	.82	
PCAP Programmer capability	1.42	1.17	1.00	.86	.70	
VEXP Virtual machine experience*	1.21	1.10	1.00	.90		
LEXP Programming language experience	1.14	1.07	1.00	.95		
Project Attributes						
MODP Use of modern programming practices	1.24	1.10	1.00	.91	.82	
TOOL Use of software tools	1.24	1.10	.91	.83		
SCED Required development schedule	1.23	1.08	1.00	1.04	1.10	

Reprinted from *Software Engineering Economics*, page 118.

*For a given software product, the underlying virtual machine is the complex of hardware and software (OS, DBMS, etc.) it calls on to accomplish its tasks.

reliability is very low, only 75% of the normal development effort (from the basic equation) will be required. On the other hand, if the required reliability is very high, 140% of the normal development effort will be required. In a similar fashion, if the analyst capability is very low, 146% of normal effort is required. Whereas, if the analyst capability will be very high, only 71% of normal development effort is required.

Question #3: How was the CER derived and what is its uncertainty?

Answer: In our example, the COCOMO CER was derived from data collected from 63 different projects, whose characteristics were available to Boehm. The total data was divided into three subsets whose data points were associated with each of the three modes, and each subset was subjected to a linear regression of the logarithms of its data points. The resulting general form of each COCOMO equation follows the common exponential form, $y = ax^b$. Comparing the actual data against the specific equations provided earlier shows that a cost estimate based on these equations will be within a factor of 1.3 of actual data only 29% of the time, and within a factor of 2 of the actuals only 60% of the time. Uncertainty is reduced by selective use of the effort multipliers detailed in Table 2. With these multipliers the cost estimate will be within a factor of 20% of the project actuals 68% of the time. Thus the Standard Error of Estimate (SEE) of COCOMO is approximately equal to 20%. It should be noted that this is not exactly true since the linear regression was made from a logarithmic transformation of the data points. A further discussion of this point will be made in a later section discussing total uncertainty of the estimate.

Question #4: What input values were used, and what is the range of uncertainty associated with each?

In any cost estimating relationship the accuracy of the output (cost or schedule) is only as accurate as the accuracy of the inputs to the CER. Thus the contractor's range of uncertainty for each input factor should be provided as a three point estimate and reviewed and validated by the government reviewing team. If there are strong differences of opinion about any value, a new range of uncertainty should be formulated by the team so that a recalculation of cost can be generated if desired.

Having selected which of the three modes of software development is most appropriate, the only other inputs to COCOMO of concern are: (1) the number of delivered source instructions (KDSI); and (2) the values of the Effort Multipliers. History has shown that estimating the value of KDSI involves high uncertainty. The final value of KDSI is invariably much higher than the original estimate since the number of lines of code is never controlled. To aid in this estimating process a software Work Breakdown Structure (WBS) should be constructed, decomposing the entire software project into software programs or modules of smaller size, (e.g. application programs, control programs, etc.). Since there is some uncertainty in the size of each, the estimated KDSI of each module should be presented as a three point estimate (most likely, optimistic, pessimistic) as in PERT analysis. Next, the set of estimates should be reviewed by a group of informed reviewers, the reasons for any large differences among the estimates should be discussed, and modifications made if the need is felt. This uses the Delphi approach to gaining consensus among the group of reviewers. The resulting final estimates which follow the series of review

discussions may be portrayed as shown in Figure 2. In this case the group consensus of the KDSI may be defined as follows: The most likely value is defined as the arithmetic mean of the (three) individual most likely values. The lowest value is defined as the minimum of all (three) minimum values. And the highest value is defined as the maximum of the (three) maximum values. While the minimum and maximum values could also have been defined as the arithmetic mean of its appropriate set of three values, the recommended method is preferred since it results in a larger range of uncertainty, making the estimate more conservative.

To obtain a probabilistic estimate of KDSI, we need to combine all of the subsystem elements obtained by consensus. This is done by first calculating the mean and the standard deviation of each element using the standard PERT formulas:

$$M = \frac{0+P+4\,ML}{6}$$

$$\sigma = \frac{P-0}{6}$$

Where: M = Mean value of each element

O = Optimistic (low) value

P = Pessimistic (high) value

ML = Most likely value

σ = Standard deviation

Finally, the mean value of KDSI for the entire system is then obtained as the sum of the mean values of all elements. The standard deviation of KDSI is obtained as the square root of the sum of the squares of the standard deviations of all elements.

In a similar fashion, the set of multipliers of Table 2 could also be reviewed by the government review team and if there is any lack of agreement, the consensus estimate, including any range of uncertainty, could be generated in the same fashion as described for KDSI.

Question #5: How does the uncertainty in the inputs and CER effect the cost output?

Answer: Having established the three point estimate for each of the input values to the model, we should now like to calculate the mean value of the cost output as well as its range of uncertainty. This may be calculated in one of the following ways. The first way is through the use of Monte Carlo simulation. Using the three point estimates, assume a probability distribution for each of the input values. Generally a Beta distribution is assumed. Insert these distributions into a Monte Carlo simulation model and run the model a large number of times, taking random draws from the set of inputs. The set of results will then constitute a probability distribution of the development effort (or duration), as shown in Figure 3. From this, the probability of the cost being less then some cost (C) may be obtained.

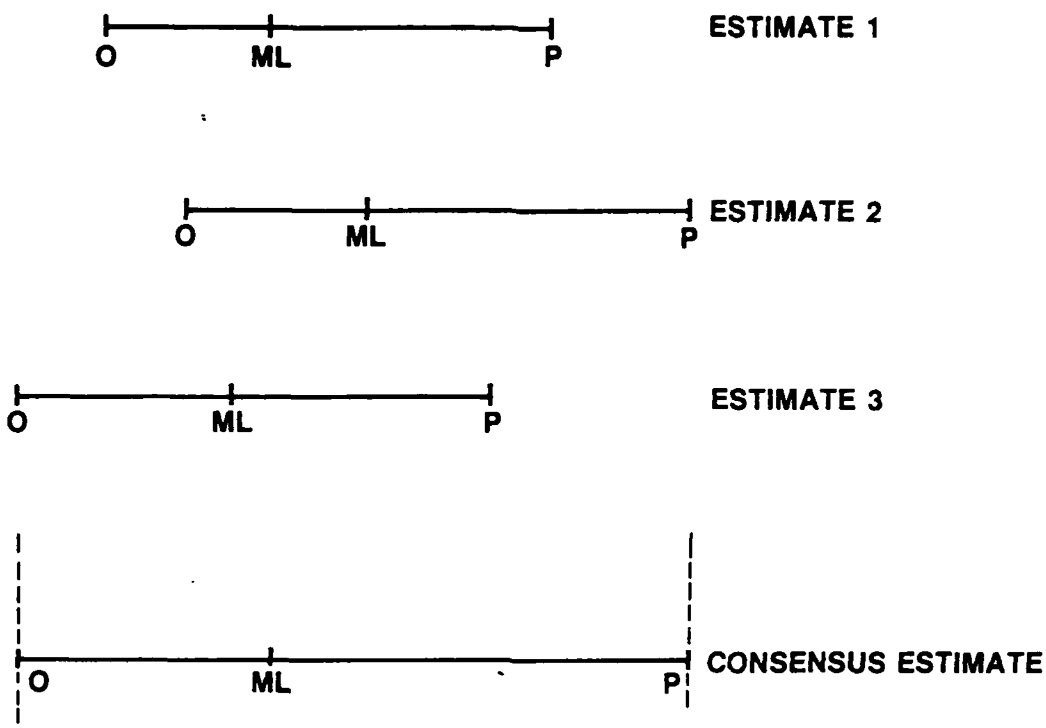


FIGURE 2. FINDING CONSENSUS ESTIMATE

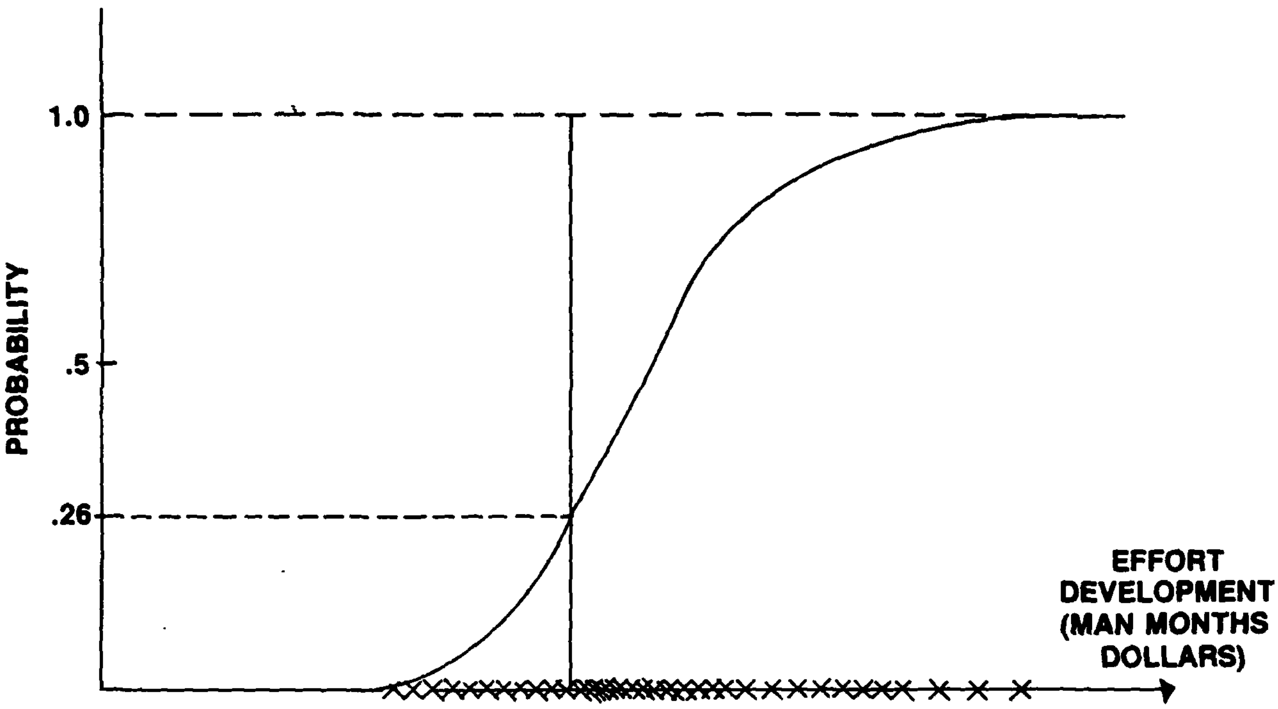


FIGURE 3. RESULTS OF MONTE CARLO SIMULATION

A second method sometimes used employs the Method of Moments, in which the output probability distribution may be obtained analytically. Unfortunately, both of these methods requires some effort in getting accurate results.

A third method can also provide a three point estimate of cost, as illustrated in Figure 4A and 4B. By inserting the mean value of the lines of code (M) and the mean values of the effort multipliers into the cost equation the mean value of cost is obtained. By repeating the calculation process using the lowest value of each input we can obtain the low value of the output. By repeating this calculation for the high values of inputs we can obtain the high value of the output estimate. These values are illustrated in Figure 4B. This type of analysis is called a sensitivity analysis and provides the PM with the best estimate (the mean value) as well as the range of uncertainty.

The range of uncertain of Figure 4B can be further refined by converting this range into a probability distribution. This is done assuming that the output cost distribution can be approximated by a Normal Probability Distribution whose mean is equal to the mean of the output estimate, and that the difference between the high and low output values (P and O) are approximately equal to six standard deviations of the normal distribution. Thus, conceptually the cost estimate can be represented as a normal probability distribution note that as shown in Figure 4C. O and P are symmetrical around the mean as long as we realize that this cost distribution is plotted on a logarithmic scale, as was mentioned previously in describing the linear regression analysis of the logarithms of the data points.

Finally, we still need to including the uncertainty of the CER itself; (recall that the accuracy of the model is within 20%, 68% of the time). This factor may be included by considering a second normal probability distribution whose mean is the same as the output mean, but whose standard deviation is 20% of the mean as illustrated in Figure 4D. Finally obtain the total impact of both uncertainties as a third new normal probability distribution whose mean value is the same as before, but whose variance is the sum of the variances of the two previous distributions. That is, the standard deviation of the final distribution is the square root of the sum of the squares of the two standards deviations.

By making the assumption that the final cost can be represented by this normal probability distribution of Figure 4D, we can now use this data to perform a Risk Analysis. Consider the example shown in Figure 5 in which the mean cost is \$100M. Also assume that the result of the previous probabilistic analysis is that the cost distribution is accurate to within a factor of 25%, 68% of the time. This means that plus one standard deviation is located at 1.25 (100) = \$125M, and minus one standard deviation is located at 100/1.25 = \$80M. If there is only \$75M in the budget, we can calculate the probability of cost overrun (or success) by using the so-called "Z table" (Table 3) associated with the standard normal probability distribution. Table 3 provides the area under the left hand "tail" of a standard normal distribution for any value of "Z", standard deviations to the left of the mean. Thus if Z were equal to 1.0 standard deviation (\$80M), the tail area could be equal to 0.1587.

In our example of Figure 5, the probability of not overrunning (i.e.,final cost being less than \$75M) is equal to the area under the curve as shown. In this case, Z has an

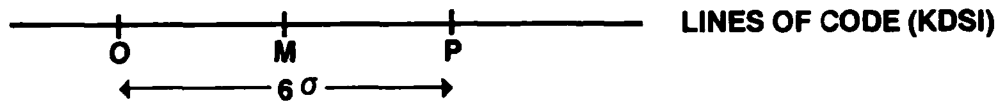


FIGURE 4A. RANGE OF UNCERTAINTY IN LINES OF CODE

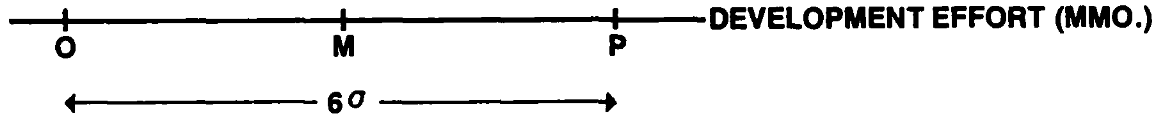


FIGURE 4B. RANGE OF UNCERTAINTY IN DEVELOPMENT EFFORT

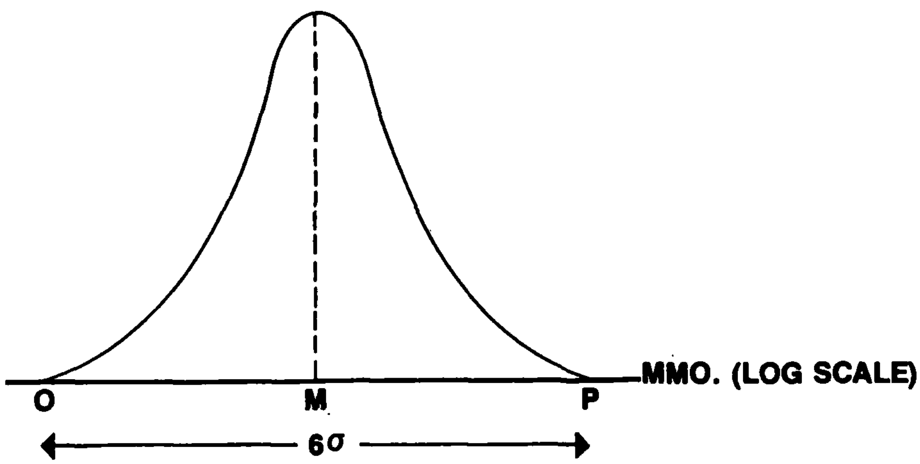


FIGURE 4C. DEVELOPMENT EFFORT AS A PROBABILITY DISTRIBUTION

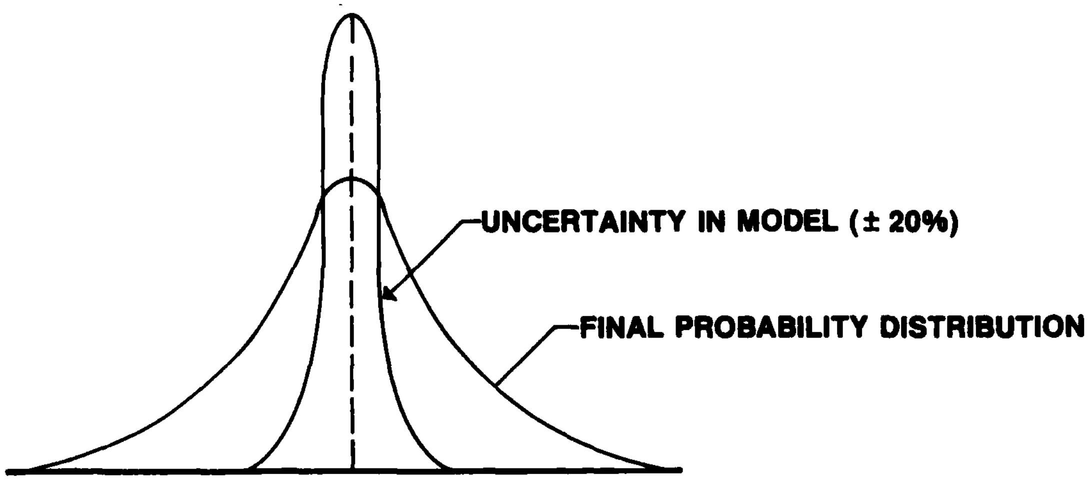


FIGURE 4D. DRIVING THE FINAL PROBABILITY DISTRIBUTION OF DEVELOPMENT EFFORT

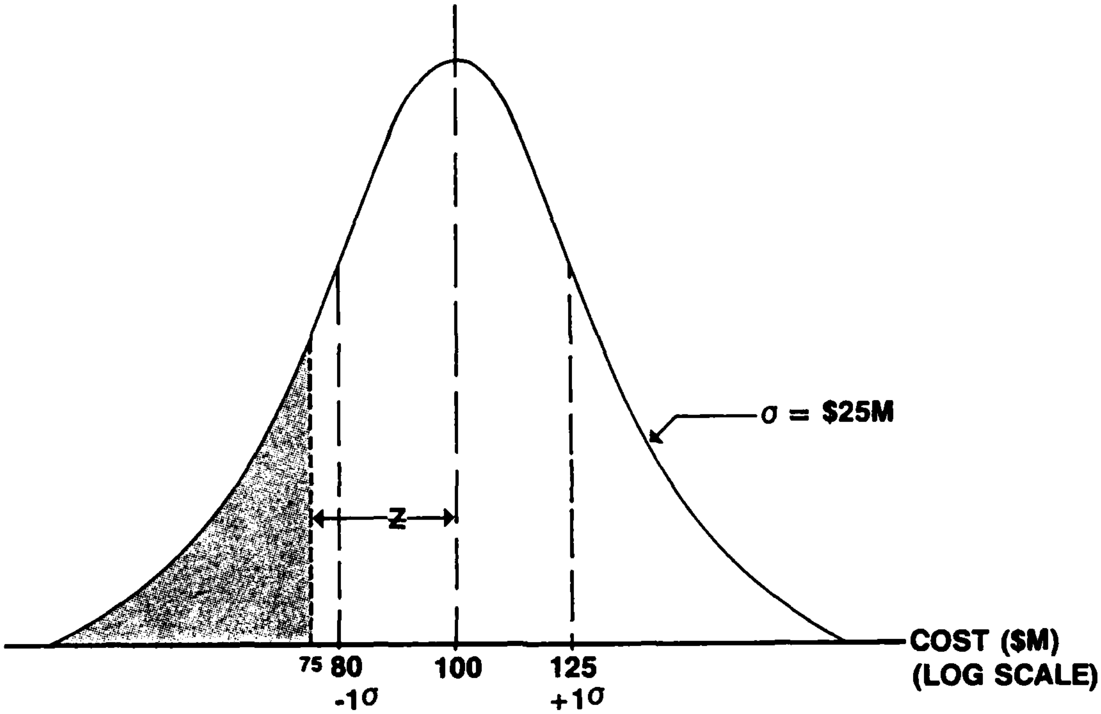
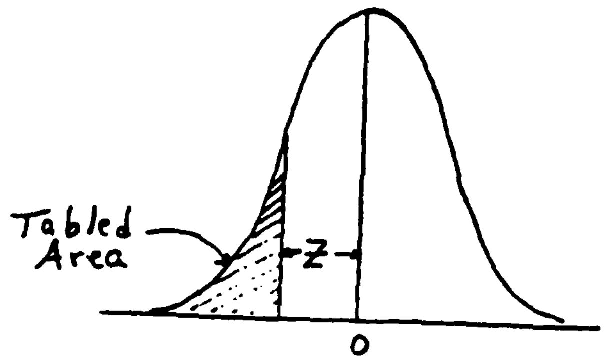


FIGURE 5. COST DISTRIBUTION

Table 3 Standard Normal Probability Distribution



AREAS UNDER THE NORMAL CURVE

Normal Deviate Z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.4960	.4920	.4880	.4840	.4801	.4761	.4721	.4681	.4641
0.1	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
0.2	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
0.3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
0.4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
0.5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
0.6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
0.7	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
0.8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
0.9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.0985
1.3	.0968	.0951	.0934	.0918	.0901	.0885	.0869	.0853	.0838	.0823
1.4	.0808	.0793	.0778	.0764	.0749	.0735	.0721	.0708	.0694	.0681
1.5	.0668	.0655	.0643	.0630	.0618	.0606	.0594	.0582	.0571	.0559
1.6	.0548	.0537	.0526	.0516	.0505	.0495	.0485	.0475	.0465	.0455
1.7	.0446	.0436	.0427	.0418	.0409	.0401	.0392	.0384	.0375	.0367
1.8	.0359	.0351	.0344	.0336	.0329	.0322	.0314	.0307	.0301	.0294
1.9	.0287	.0281	.0274	.0268	.0262	.0256	.0250	.0244	.0239	.0233
2.0	.0228	.0222	.0217	.0212	.0207	.0202	.0197	.0192	.0188	.0183
2.1	.0179	.0174	.0170	.0166	.0162	.0158	.0154	.0150	.0146	.0143
2.2	.0139	.0136	.0132	.0129	.0125	.0122	.0119	.0116	.0113	.0110
2.3	.0107	.0104	.0102	.0099	.0096	.0094	.0091	.0089	.0087	.0084
2.4	.0082	.0080	.0078	.0075	.0073	.0071	.0069	.0068	.0066	.0064
2.5	.0062	.0060	.0059	.0057	.0055	.0054	.0052	.0051	.0049	.0048
2.6	.0047	.0045	.0044	.0043	.0041	.0040	.0039	.0038	.0037	.0036
2.7	.0035	.0034	.0033	.0032	.0031	.0030	.0029	.0028	.0027	.0026
2.8	.0026	.0025	.0024	.0023	.0023	.0022	.0021	.0021	.0020	.0019
2.9	.0019	.0018	.0018	.0017	.0016	.0016	.0015	.0015	.0014	.0014
3.0	.0013	.0013	.0013	.0012	.0012	.0011	.0011	.0011	.0010	.0010

Reprinted from Statistics for Business Decisions, by Ernest Kurnow, Gerald Glasser, and Fred Ottman, published by Richard D. Irwin, Inc., 1959.

absolute value greater than one standard deviation. Ordinarily the cost distribution would be plotted on a linear scale, and Z would equal the ratio of the deviation from the mean to the standard deviation 25/20 = 1.25. However, since the cost axis is drawn on a log scale (so that the cost distribution will have the shape of normal probability distribution) we must calculate Z as a ratio of logarithms.
Thus Z = (log 25)/(log 20) = 1.398/1.301 = 1.074 standard deviations from the mean. Thus from Table 3, there is a 14% chance of success (equivalent to a 86% chance of cost overrun).

Question #6: How well does the model correspond to this particular project, or how can it be made to correspond?

Answer: All cost estimating models consist of some form of extrapolation of some set of relevant historical data. When we audit someone else's cost estimate, we should check the relevance of the data base used by determining if the cost estimator generated his own CER from his own data (through some regression analysis), or if he used someone else's CER. If the former case is true, the auditor can determine how similar the project being estimated is to the projects from which the data used in the regression.

In our COCOMO example the regression analysis was based on a set of 63 past aerospace projects whose characteristics were available to Boehm and were representative of the way that TRW develops software. Hence Boehm's equations represent the way that TRW development teams may perform in the future. However, if another company is involved, their cost estimate should be different since their development work process and their method of counting KDIS, man-months, etc. may differ from that of TRW. In this case the cost estimator needs to "fine tune" the COCOMO model to correspond to the way his organization operates. We call this fine-tuning "calibrating" or "tailoring" the model. Calibration is done as follows: First identify several (the more the better) software development projects which the estimator's organization has completed in the past. These projects should be of the same type as the new project. Second, gather data on the actual number of delivered lines of code (KDIS), development effort and time, and values of the input multipliers for each of the past projects. Next, insert this input data for each project into the COCOMO model and calculate the mean value of the cost estimate for each project. Compare the COCOMO cost estimate to the actual completed value. Suppose we find that on average (an average weighted by the lines of code in each project) the true result is 12% greater that the COCOMO estimate. One way of adjusting or calibrating the COCOMO model to the contractor's method of operation is by including an additional multiplier factor, in this case, Mc = 1.12. A more scientific method is to use a least-squares approximation technique to calibrate the constant term for the development mode equation to the organization's project data. This technique is described in detail in Boehm's book.

CONCLUSIONS

One method that a government reviewer can use in validating a contractor's cost estimate is to motivate the contractor to generate the cost estimate properly, and to forward the entire analysis, including all of the data used, to the government for review. If the analysis has been done properly, the government's effort is reduced to one of checking, rather than independent cost estimating. Furthermore, if several contractors are providing estimates, the government review team can check

corresponding parts of the analysis against one another, draw their own conclusions of what the CER input values should be (including the range of uncertainty of each input characteristic), and obtain a good bounded range of the estimated cost. This paper describes and illustrates the type of instructions which can be given to the contractors to provide such motivations, and how the government can use the contractor's data to obtain its own estimate.

REFERENCES:

Boehm, Barry W., Software Engineering Economics, Prentice-Hall, Inc., 1981.

Rudwick, Bernard H., Adjusting Software Estimating Models to Obtain Comparative Results, Journal of Parametrics, March 1984.

END

FILMED

1-86

DTIC